

Computing Power Scheduling Platform and Computing Power Leasing System Thereof

Technical Field

[0001] Embodiments of the present invention relate to the technical field of artificial intelligence, and in particular, relate to a computing power scheduling platform and a computing power leasing system thereof.

Background

[0002] With the continuous development of artificial intelligence technology, demands for computing power resources from many individuals and enterprise organizations are increasing day by day. Due to the high cost of direct deployment and purchase of physical devices, leasing service for computing power resources has emerged to provide individuals and organizations with relatively affordable and accessible computing power resources. Typical computing power leasing business mainly includes bare-metal leasing and leasing of machine equipped with NVIDIA basic system platform.

[0003] Bare-metal leasing refers to only providing physical devices without any system or software, and the system and software are installed by users themselves. In this application scenario, users must possess the capability to deploy from the underlying system. Moreover, the physical device can only be leased as a whole (usually 8 CPU cards per device), and leasing of a single operation card is not available.

[0004] Therefore, when the computing power demand for the actual task does not exceed 8 CPU cards, it will cause certain computing power waste. Moreover, the artificial intelligence development environment also needs to be built by users, which is time-consuming and challenging. If the operation of the platform only relies on operation and maintenance personnel, then the difficulty and workload of operation and maintenance become significantly high.

[0005] The leasing service for machine equipped with NVIDIA basic system platform means that the server is equipped with the NVIDIA basic system platform. Users can use computing power by adapting to the system interface. In this application scenario, users do not need the capability of deploying from the underlying system, but only need to adapt to NVIDIA related interfaces.

[0006] Similarly, such a leasing service also can only provide whole-machine leasing, and cannot support the usage-based billing mode. Moreover, the AI development environment also needs to be built by users, which is time-consuming and challenging. If the operation of the platform only relies on

operation and maintenance personnel, then the difficulty and workload of operation and maintenance become significantly high.

[0007] Therefore, there is an urgent need to provide an appropriate method to solve the problems of low efficiency of computing power utilization and inconvenient management and use in the existing computing power leasing service.

Summary of the Invention

[0008] Embodiments of the present application aim to provide a computing power scheduling platform and a computing power leasing system thereof, which can solve at least part of the defects of the existing computing power leasing service.

[0009] In a first aspect, the present application provides a computing power scheduling platform. The computing power scheduling platform comprises: an order management unit, being configured to receive at least one computing task from a platform user; a storage space allocation unit, being configured to allocate a data storage space with a target capacity for the platform user; a computing power allocation unit, being configured to allocate a corresponding computing node to the computing task according to computing power requirements of the computing task; an expense calculation unit, being configured to record the usage of computing resources of the computing task and generate corresponding expense information; wherein the computing task is packaged in an independent container, and the container is launched on the computing node allocated by the computing power allocation unit so that the computing task is executed; data information formed after the computing task is executed is stored in the data storage space allocated by the storage space allocation unit.

[0010] Optionally, the computing power scheduling platform further comprises a monitoring unit which is configured to: feedback the state information of the computing node during the execution of the computing task; and the state information is selected from one or more of the following: CPU load; GPU load and physical memory usage ratio.

[0011] Optionally, the computing power scheduling platform further comprises a platform interface presented on a user terminal; wherein the platform interface comprises an access page for accessing the container.

[0012] Optionally, the platform interface further comprises a user login page for the platform user to register and log in.

[0013] Optionally, the target capacity is determined by the platform user; and the data storage space is mapped into the container.

- [0014] Optionally, the data storage space is provided with a plurality of storage space copies.
- [0015] Optionally, the data information formed after the computing task is executed is stored based on a persistent storage mechanism so that the data information is independent of the life cycle of the container.
- [0016] Optionally, the computing power allocation unit is specifically configured to: determine the computing power requirements of the computing task; authorize a computing node meeting the computing power requirements to a corresponding platform user, and binding the authorized computing node with the platform user.
- [0017] In a second aspect, an embodiment of the present application further provides a computing power leasing system. The system comprises: a physical device providing computing power resources and storage spaces; and the computing power scheduling platform as described above; wherein the computing power scheduling platform abstracts the computing power resources and the storage spaces of the physical device through the container and virtualization technology to form corresponding virtual resources.
- [0018] Optionally, the computing power scheduling platform forms a virtual resource corresponding to the physical device through a Docker container and a KVM virtualization platform.
- [0019] Different from the situation in the prior art, the embodiments of the present invention realize unified scheduling and management of computing power resources through the container and virtualization technology, so that users can conveniently use computing power resources under platform management and obtain corresponding calculation results. Moreover, the leasing expense can be calculated according to the amount of computing power used, which effectively improves the utilization rate of computing power and reduces the use cost of customers, thereby avoiding unnecessary computing power waste.

Description of the Drawings

- [0020] One or more embodiments are exemplarily described with reference to corresponding attached drawings, and these exemplary descriptions are not intended to limit the embodiments. In the attached drawings, elements with the same reference numerals represent the similar elements, and unless otherwise stated, the pictures in the attached drawings are not intended to limit the scale.
- [0021] FIG. 1 is an application scenario of a computing power scheduling platform according to an embodiment of the present application.

- [0022] FIG. 2 is a schematic view of the data processing flow of the computing power scheduling platform according to the embodiment of the present application.
- [0023] FIG. 3 is a functional block diagram of the computing power scheduling platform according to the embodiment of the present application.
- [0024] FIG. 4 is a page diagram of the computing power scheduling platform according to the embodiment of the present application.
- [0025] FIG. 5 is a schematic view of a computing power leasing system according to an embodiment of the present application.

Detailed Description of the Embodiments

- [0026] In order to facilitate the understanding of the present application, the present application will be described in more detail hereinafter with reference to the attached drawings and specific embodiments. It shall be noted that when an element is said to be “fixed” to another element, it may be directly located on the other element, or there may be one or more intervening elements therebetween. When an element is said to be “connected” to another element, it may be directly connected to the other element, or there may be one or more intervening elements therebetween. Orientation or positional relationships indicated by terms such as “upper”, “lower”, “inner”, “outer” and “bottom” used in this specification are orientation or positional relationships shown based on the attached drawings. These terms are only used for the convenience of describing the present application and simplifying the description, and do not indicate or imply that the devices or elements referred to must have a specific orientation, or must be constructed and operated in a specific orientation, and thus they should not be understood as limitations to the present application. In addition, terms such as “first”, “second” and “third” are only used for descriptive purposes and should not be construed as indicating or implying relative importance.
- [0027] Unless otherwise defined, all technical and scientific terms used in this specification have the same meaning as commonly understood by those skilled in the art of the present application. The terms used in the specification of the present application are only used for the purpose of describing specific embodiments, and are not intended to limit the present application. The term “and/or” as used in this specification includes any and all combinations of one or more associated items listed.

[0028] In addition, the technical features involved in different embodiments of the present application described below can be combined with each other as long as they do not conflict with each other.

[0029] The technical solutions in the present application will be described below with reference to the attached drawings.

[0030] FIG. 1 is an application scenario of a computing power scheduling platform provided according to an embodiment of the present application. As shown in FIG. 1, in this application scenario, a plurality of different user terminals 10, a computing power scheduling platform 20, a physical device 30 and a network service 40 are included.

[0031] The user terminal 10 is a terminal device through which users interact with the computer system. It may allow users to input instruction information, perform one or more processing on the instruction information and/or present feedback information formed after the processing.

[0032] Specifically, the user terminal may be any type of intelligent devices with logical operation capability, including but not limited to smart phones, personal computers, laptops and intelligent wearable devices. It shall be noted that the user terminal does not particularly refer to a single device, and it may be composed of a variety of device components which are in communication connection with each other and used to perform different functions respectively, or it is formed by integrating multiple functional modules into the same device.

[0033] The user terminal 10 may also be configured with one or more different interactive devices for collecting user instructions or displaying and feeding back information to users. These interactive devices include, but are not limited to, input keyboards, display screens, touch screens, and loudspeakers or the like.

[0034] The computing power scheduling platform 20 is a system platform that provides various computing resources and services over networks and integrates multiple functions. It can abstract resources such as computing power and storage spaces provided by the physical device 30 into virtual resources through the container and virtualization technology, thereby providing services with on-demand allocation, elastic scaling and high availability.

[0035] Specifically, the computing power scheduling platform 20 can abstract the physical device 30 through the Docker container and the KVM virtualization platform, so as to facilitate management of the physical device.

[0036] The physical device 30 is one of tangible physical devices that provide computing power resources, storage spaces and other resources. These tangible physical devices support the operation of computing systems and cloud platforms, which together constitute the computing infrastructure. It includes,

but is not limited to, a central processing unit (CPU) and a graphics processing unit (GPU) that provides computing resources, a hard disk drive (HDD) and a solid-state hard disk (SSD) that provide storage space for persistent storage of data, a power supply unit (PSU), and a cooling system or the like.

[0037] In the present application, for convenience of description, terms “first physical device” and “second physical device” are respectively used to represent device components for providing computing resources and storage spaces. However, as shall be appreciated by those skilled in the art, the “first physical device” and “second physical device” are only classified descriptions of hardware components according to different functions to be performed, and are not intended to specifically distinguish different devices. That is, the first physical device and the second physical device may be different components in the same device or they may be different devices.

[0038] The network service 40 is a wireless communication network for establishing a data transmission channel between two nodes, which is specifically based on any type of data transmission principle, such as a WiFi network in a specific signal frequency band, a wireless cellular network, or a combination thereof, so as to establish a communication connection between the user terminal 10, the computing power scheduling platform 20 and the physical device 30.

[0039] During the actual use, the user can use the user terminal 10 to log into the computing power scheduling platform 20 by logging into a portal website or launching a dedicated mobile application, and send a computing power leasing request or a similar order.

[0040] Subsequently, the computing power scheduling platform 20 allocates corresponding storage spaces and computing power resources through automatic scheduling, so that the computing power use request of the user can be fulfilled. The calculated data information and related user data may be stored in an independent and isolated storage space, which is convenient for users to view and obtain through the user terminal 10.

[0041] The computing power scheduling platform 20 provided according to the embodiment of the present application adopts the platform integrating the docker container and KVM virtualization, and eliminates the need for agent program installation on computing nodes, thereby realizing unified management of the whole network and reduce the use difficulty. Customers or users can conveniently place orders by themselves through the user terminal 10 for scheduling and allocation of computing resources, thereby realizing unified management of orders, workflows and billing, in compatibility with Nvidia DGX, Ubuntu, RHEL and other operating systems.

- [0042] The computing power scheduling platform 20 defines partition and authorization, and realizes fine-grained partition isolation and license management. It can allocate and authorize corresponding computing resources to users according to the computing task, so as to realize authorized access spanning from bare metal to individual containers, and ensure user data security.
- [0043] It should be noted that the application scenario shown in FIG. 1 is only used for the illustrative purpose. One or more devices may be added or omitted by those skilled in the art according to the actual needs, without being limited to those shown in FIG. 1.
- [0044] FIG. 2 is a schematic view of the data processing flow of the computing power scheduling platform provided according to the embodiment of the present application. As shown in FIG. 2, the data processing method includes the following steps:
- [0045] S100. submitting a computing task through a user terminal by a platform user.
- [0046] The “platform user” refers to an account that has registered relevant user information on the computing power scheduling platform and completed the registration. It may be registered and generated in advance by users or customers on portal websites or specific mobile applications through user terminals, and account management is carried out by using any existing and suitable platform user management mode.
- [0047] S200. verifying the platform user and the computing task submitted from the platform user by the computing power scheduling platform. Upon successful verification, the computing task is bound with the corresponding platform user and authorized.
- [0048] The verification can be achieved by any suitable means, including but not limited to account passwords and biometric fingerprint information or the like. Through the authority management of verification, unauthorized access can be prevented to ensure the security and privacy of data.
- [0049] S300. allocating a data storage space with a target capacity for the platform user among the storage resources of the distributed storage architecture.
- [0050] The target capacity may be determined by the customers or users according to actual needs, and no specific limitation is made thereto.
- [0051] In addition, the data storage space is mapped into the container and used to store the data information of the platform user. Specifically, the mapping of the data storage space may be realized by means of Volumes or Bind Mounts.

- [0052] Preferably, the data storage space may adopt the distributed redundant storage architecture, in which multiple storage space copies are provided. These storage space copies store data such as user models, mirror images, calculation and settlement, which may be stored on different physical nodes respectively. Therefore, when the computing node currently being used fails, the mirror image can be switched to the normal computing node in seconds, so as to solve the problems that the task cannot be executed and the data is lost due to the failure of the computing node, and ensure that when a hardware failure occurs to the computing node, the computing task can be quickly switched to other normally functioning computing nodes to continue the execution of the computing task.
- [0053] S400. allocating and scheduling the corresponding computing nodes by the computing power scheduling platform according to the computing power demand of the computing task so that the computing task can be fulfilled.
- [0054] Each computing task is packaged in a separate container. Therefore, with the initiation of the container on the computing node, the computing task can be executed on the corresponding computing node.
- [0055] In the present application, the term “container” is used to represent a lightweight and portable virtualization technology. It encapsulates the application along with all its dependencies so that the application can run consistently in any environment that supports container runtime. The term “computing node” is used to refer to a physical device or a virtual machine that provides actual computing resources. It may be considered as the basic unit of the computing power scheduling platform for performing various computing tasks.
- [0056] The term “authorized computing node” refers to a computing node that is allocated to a specific platform user. The authorized computing node may be bound with the corresponding platform user, so that the logged-in platform users can access and control these computing nodes by accessing the access page of the container.
- [0057] Specifically, based on different computing power resources required by the computing tasks, the authorized computing nodes may also have corresponding different hierarchical tiers. For example, for computing tasks that require more computing power resources, a single device may be authorized, while for computing tasks that require less computing power resources, a single computing card in a single device may be authorized. Further speaking, for a computing task requiring very little computing power resources, a certain container may be authorized. As an example, but not a limitation, the computing

power scheduling platform adopts a predictive resource scheduling algorithm, which can preset computing resources in advance based on historical task execution data, current system load and time pattern, and reduce the waiting time of users. This algorithm can identify a periodic task pattern, and actively adjust the resource allocation strategy before the peak period to ensure the balanced utilization of resources.

[0058] S500. collecting the specific usage of computing power and calculating the corresponding computing power leasing expense according to the collected usage of computing power by the computing power scheduling platform as the computing task begins to be executed.

[0059] The computing power leasing expense may be calculated according to a preset billing rule. The specific billing rule may be determined according to actual needs, and no specific limitation is made thereto. Preferably, the billing rule may also be an adjustable rule, which may be adjusted correspondingly in response to different leasing demands. For example, specific discounts are provided for long-term and stable computing power leasing demands.

[0060] S600. providing the computing nodes allocated and scheduled by the computing power scheduling platform to the user terminal through the corresponding container access page, so that the user can conveniently operate and control the corresponding computing node.

[0061] S800: storing the data information generated after calculation of the computing node in the data storage space so as to be read in response to the request of the user terminal.

[0062] The data information formed after the computing task is executed is stored based on a persistent storage mechanism, so that these data information are independent of the life cycle of the container and can still be accessed and acquired by the user terminal 10 even after the container is destroyed.

[0063] In some embodiments, in order to facilitate the customer or user to know the running condition of the computing node in time, the computing power scheduling platform also provides the function of task monitoring. Continuing to refer to FIG. 2, the corresponding data processing method also includes:

[0064] S700. feeding back the state information of the computing node in real time in the process of executing the computing task.

[0065] The state information is selected from one or more of the following: CPU load; GPU load and physical memory usage ratio. The state information may be displayed and presented in any suitable way according to the actual needs. For example, as shown in FIG. 4, the number of computing nodes, the number of available and utilized cores of CPU, the number of available and utilized GPUs,

the available and utilized capacity of memory or the like may be displayed respectively. Based on the data processing flow provided in the above embodiment, the embodiment of the present application further provides the concrete structure of the computing power scheduling platform. FIG. 3 shows a functional block diagram of the computing power scheduling platform 20 according to the embodiment of the present application.

[0066] As shown in FIG. 3, the computing power scheduling platform 20 includes an order management unit 21, a storage space allocation unit 22, a computing power allocation unit 23 and an expense calculation unit 24.

[0067] The order management unit 21 is configured to receive at least one computing task from a platform user. The storage space allocation unit 22 is configured to allocate a data storage space with a target capacity for the platform user. The computing power allocation unit 23 is configured to allocate a corresponding computing node to the computing task according to the computing power requirements of the computing task. The expense calculation unit 24 is configured to record the usage of the computing resources of the computing task and generate corresponding expense information.

[0068] During the actual use, the platform user submits his/her leasing order through the user terminal, and specifies the required computing tasks and/or computing power requirements. The computing power scheduling platform automatically selects and allocates appropriate resources (e.g., computing nodes and storage spaces) for the user through automatic scheduling. Subsequently, the computing power scheduling platform binds the resources determined by the allocation with the identity of the platform user to ensure that only the platform user can access and use these resources.

[0069] When the platform users instruct the computing tasks to run, the computing power scheduling platform packages these computing tasks into containers, and launches the containers on the allocated computing nodes to start the execution of the tasks. At the same time, with the execution of the computing tasks, the computing power scheduling platform also begins to record the usage of computing power and calculate the corresponding leasing expense.

[0070] In some embodiments, the order management unit also has a task priority dynamic adjustment mechanism, which can intelligently adjust the task execution priority according to factors such as user level, task urgency degree, system load status, etc., thereby ensuring that key tasks can be given priority in the case of limited resources, while maximizing the overall throughput of the system.

- [0071] In some embodiments, still referring to FIG. 2, the computing power scheduling platform 20 may further include a monitoring unit 25. The monitoring unit 25 is configured to feed back the status information of the computing node during the execution of the computing task. The state information is selected from one or more of the following: CPU load; GPU load and physical memory usage ratio.
- [0072] Therefore, users or customers can conveniently check the current execution of computing tasks, the usage of resources, and the leasing expense information through portals supported by the computing power scheduling platform.
- [0073] Preferably, the computing power scheduling platform may also provide the function of automatic fault alarm. It can automatically give an early warning before the failure (e.g., when a specific parameter exceeds the threshold), so that the operation and maintenance personnel can predict the fault point according to the parameter situation and deal with it pertinently to save the inspection time.
- [0074] Moreover, during the fault handling, the computing task can be conveniently dispatched to other physical devices, so that the platform user can continue to use the service without waiting for fault resolution. After the fault is resolved, the physical device can be reintegrated into the abstracted virtual resource pool and wait for scheduling for other computing tasks.
- [0075] In addition, the monitoring unit 25 is not only responsible for feeding back the status information of the computing node, but also for mining the resource usage pattern. Through the analysis of historical monitoring data, the computing power scheduling platform 20 can identify the time pattern, task type pattern and user behavior pattern of resource utilization, and provide data support for predictive resource allocation. The data collected by the monitoring unit 25 will be used as the training set of the machine learning model to continuously optimize the computing power scheduling strategy.
- [0076] The computing power allocation unit 23 is further configured to predict the resource demand feature of the computing task based on the machine learning model, which is obtained by training historical task execution data. The computing power scheduling platform 20 can intelligently match the most suitable computing node according to multidimensional features such as the computing task type, input data scale and expected execution time, rather than simply judging resource availability. The computing power allocation unit 23 continuously learns the relationships between user task features and resource utilization patterns to continuously optimize the allocation strategy.

- [0077] Specifically, the computing power allocation unit 23 calculates the optimal resource allocation scheme by the computing power scheduling platform based on the trained task-resource matching model according to the computing power demand feature vector of the computing task, and schedules and allocates the corresponding computing nodes. The task-resource matching model can comprehensively consider the computing-intensive, memory-intensive, or IO-intensive and other attributes of computing tasks, as well as factors such as node load balance and energy efficiency, so as to achieve the global optimal resource allocation.
- [0078] The computing power demand feature vector is a multidimensional numerical representation of computing task resource demand, which quantifies the computing characteristics of tasks into structured data. This vector usually includes key dimensions such as computational intensity, memory requirement, I/O intensity, parallelism, GPU acceleration adaptability, expected execution time, data scale and task type identification. For example, the feature vectors of deep learning training tasks may be characterized by a feature combination of high computational intensity, high GPU adaptability and long execution time.
- [0079] The task-resource matching model is a trained machine learning model, which establishes a mapping function from task features to optimal resource allocation by analyzing the relationships between the feature vectors of historical tasks and the optimal resource allocation.
- [0080] The core of the task-resource matching model lies in its data-driven training process, and different from the traditional rule configuration system, this model is built and improved through continuous training of large-scale historical execution data. The training data set contains thousands of complete life cycle records of historical computing tasks, and each record includes multidimensional indices such as task feature vector, distributed resource allocation, actual execution time, resource utilization rate and energy efficiency. The system will automatically mark the optimal resource configuration, i.e., the configuration with the highest resource utilization rate under the premise of meeting the performance requirements, as the target label of training.
- [0081] The model training process adopts a multi-stage deep learning architecture, including three core components: feature extraction, pattern recognition and decision generation. In the feature extraction stage, the system uses the convolutional neural network to process the task code structure and data distribution feature. In the pattern recognition stage, the recurrent neural network analyzes the time sequence pattern of resource utilization in the process of task execution. In the decision generation stage, the reinforcement

learning algorithm optimizes the resource allocation strategy to balance immediate performance with long-term system efficiency.

[0082] With the increase of accumulated training data, the task-resource matching model is more accurate in understanding the resource demand features of various tasks. For example, for deep learning training tasks, the task-resource matching model has been able to predict the GPU memory consumption curve from information such as the network architecture, batch size and optimizer selection. For distributed data processing tasks, the task-resource matching model can accurately estimate the network bandwidth requirements based on the data sharding strategy and aggregation pattern.

[0083] In practical application, with the increase of the number of tasks handled by the computing power scheduling platform, the prediction accuracy of the task-resource matching model shows a logarithmic growth trend, and the accuracy is greatly improved and tends to be stable after a lot of training. The intelligent matching model based on massive data training has completely changed the traditional paradigm of computing resource allocation, and evolved from static rules to dynamic learning, which has brought significant technical breakthroughs and efficiency improvements to the computing power scheduling system.

[0084] There is a close input and reasoning relationship between the computing power demand feature vector and the task-resource matching model. As the core input of the model, the feature vector is received and processed by the task-resource matching model, and finally the optimal resource allocation scheme is output. The essence of this process is to transform abstract features of tasks into specific resource allocation decisions. The feature vector provides a mathematical abstract representation of tasks, while task-resource matching model learns the complex mapping relationships from feature space to resource allocation space through a large number of historical samples, and realizes the intelligent transformation from task description to resource allocation.

[0085] Taking deep learning training tasks as an example, the computing power scheduling platform will first analyze the submitted tasks, extract features such as network structure complexity, batch size, data set scale, GPU affinity and expected training duration, and form feature vectors to input into the task-resource matching model. Based on the historical learning pattern, the task-resource matching model may predict the optimal resource allocation that requires 4 high-performance GPUs, at least 32GB of video memory, 128GB of system memory and high-speed NVMe storage. The computing power scheduling platform then allocates corresponding computing nodes according to

this decision, monitors the implementation, and feeds back the implementation results for further optimization.

[0086] In some other embodiments, the computing power allocation unit 23 also has an adaptive resource elastic scaling mechanism, which can dynamically adjust the allocated computing resources according to the actual resource usage during the execution of computing tasks. When it is detected that the resource requirements of computing tasks have changed, the allocated computing resources can be increased or decreased in real time to ensure the maximum utilization efficiency of resources and meanwhile meet the performance requirements of tasks.

[0087] In some other embodiments of the present application, the computing power scheduling platform 20 further includes a task analysis unit which is configured to perform feature extraction and analysis on computing tasks submitted by users, including but not limited to task type identification, resource consumption pattern prediction, execution duration estimation or the like. The task analysis unit adopts a deep learning model, which can learn from task codes, data scale and historical execution records, and provide decision-making basis for the computing power allocation unit 23.

[0088] As shall be clearly appreciated by those skilled in the art, for the convenience and conciseness of description, reference may be made to the corresponding processes in the aforementioned method embodiments for the specific working processes of the devices and modules described above, and this will not be further described herein. Those of ordinary skill in the art can recognize that the units and algorithm steps of the respective examples described in combination with the embodiments disclosed herein can be realized by electronic hardware, computer software or a combination thereof. In order to clearly illustrate the interchangeability of hardware and software, the composition and steps of the respective examples have been generally described in terms of functions in the above description. Whether these functions are implemented in hardware or software depends on the specific application and design constraints of the technical scheme. Professionals can use different methods to implement the described functions for each specific application, but such implementation should not be considered beyond the scope of the present application.

[0089] Those skilled in the art can use different methods to implement the described functions for each specific application, but such implementation should not be considered beyond the scope of the present application. The computer software may be stored in a computer-readable storage medium, and

when the program is executed, it may include the process flow of the embodiments of the above methods. The storage medium may be a magnetic disk, an optical disk, a read-only memory or a random access memory or the like.

[0090] To sum up, the computing power scheduling platform provided according to the embodiment of the present application has the following features.

[0091] 1) Intelligent resource matching: the computing power scheduling platform can accurately understand the resource demand features of computing tasks through the data-driven task-resource matching model. The model is trained based on massive historical execution data and adopts multi-stage deep learning architecture, including components of feature extraction, pattern recognition and decision generation. It can not only learn resource requirements from explicit features, but also identify implicit calculation patterns to predict resource consumption curves. With the increase of the number of tasks processed, the accuracy is greatly improved and tends to be stable after a lot of training, and a technical breakthrough from static rules to dynamic learning is realized.

[0092] 2) Improvements on utilization rate of computing power: based on the training-optimized intelligent scheduling model, the computing power scheduling platform can deeply analyze the matching degree between the current load of the computing power server and the task feature to realize the optimal task allocation. The multi-objective optimization framework takes execution efficiency, resource utilization rate and energy consumption cost into consideration at the same time, which increases the utilization rate of computing power from a maximum of 30% in traditional methods to over 80%, showing an improvement of more than 167%. Through the online learning mechanism, the prediction results are continuously compared with the actual execution data, so that the resource allocation strategy is continuously optimized.

[0093] 3) Accurate resource customization: through feature vector extraction and model prediction, users do not need to specify detailed resource configuration manually, and the computing power scheduling platform can automatically recommend the most suitable resource combination according to task characteristics. Users can get resource allocation accurate to the level of individual computing cards, and realize the transformation from "whole machine leasing" to "on-demand allocation". The model can intelligently recommend the most economical and efficient resource allocation according to the unique

resource consumption patterns of different task types (such as deep learning training, reasoning, data analysis, etc.).

[0094] 4) Optimized user experience: in the system based on the computing power scheduling platform, users only need to submit computing tasks, and they can obtain optimal resource allocation without professional knowledge. The computing power scheduling platform provides an open source suite for AI development and an adaptive running environment, and the system will analyze the task features to automatically match the most suitable development environment. The transfer learning mechanism can identify new types of tasks and adapt quickly to ensure that demands of all kinds of users can be satisfied, which saves more than 1 to 2 days of environment configuration time.

[0095] 5) Dynamic and fine billing: the pay-as-you-go billing mode of the computing power scheduling platform is based on the real-time monitoring of resource usage and predictive models, which can accurately allocate resources when the computing task is started and avoid the cost waste caused by over allocation. Through continuous learning of user task patterns, the resource consumption curve of computing tasks can be predicted, and dynamic resource adjustment and fine billing based on the consumption curve can be realized, with accuracy down to the minute level.

[0096] 6) Intelligent early warning and self-healing: the computing power scheduling platform integrates an anomaly detection algorithm based on machine learning, and establishes a predictive maintenance model by analyzing historical performance data and failure pattern. The computing power scheduling platform implements 24-hour intelligent monitoring of bare metal, computing power, services and key processes, so that it can predict potential failures and issue early warnings. The reinforcement learning algorithm optimizes the fault response strategy, automatically performs resource migration and recovery operations, minimizes the need for manual intervention, significantly improves the efficiency of operation and maintenance, and simplifies and highly automates the operation and maintenance work.

[0097] Different from the prior art, the computing power scheduling platform provided according to the embodiment of the present application can provide intelligent resource allocation far beyond simple rule matching through data training model matching, and realize accurate matching of resources and tasks. This data-driven decision-making method can dynamically adapt to the resource demand characteristics of different types of tasks, maximize the system resource utilization rate while ensuring the task performance, greatly improve the

calculation efficiency and reduce the cost, and provide users with better computing service experience.

[0098] FIG. 5 shows a schematic structural diagram of a computing power leasing system according to an embodiment of the present application. The specific implementation of the computing power leasing system is not limited by the specific embodiment of the present application, and it may be a hardware device for implementing the above computing power scheduling platform or a physical device as shown in FIG. 1.

[0099] As shown in FIG. 5, the computing power leasing system may include a processor, a communication interface, a memory and a communication bus.

[0100] The processor, the communication interface and the memory communicate with each other through the communication bus. The communication interface is used for communicating with network elements of other devices, such as clients or other servers. The processor is used to execute the program, and specifically, it can execute one or more steps in the above method embodiment (for example, one or more steps in the above data processing flow).

[0101] Specifically, the program may include program codes, which include computer operation instructions. In the embodiment of the present application, according to the type of hardware used, the processor may be a Central Processing Unit (CPU), and the processor may also be other general-purpose processors, Digital Signal Processors (DSPs), Application Specific Integrated Circuit (ASIC), Field-Programmable Gate Array (FPGA) or other programmable logic devices, discrete gate or transistor logic devices, discrete hardware components or the like.

[0102] The memory is used to store programs. The memory 1306 may include a high-speed RAM memory, and it may also include a non-volatile memory, such as at least one disk memory. The program may be specifically used to cause the processor 1302 to execute one or more steps in any of the above method embodiments.

[0103] Finally it shall be noted that, the above embodiments are only used to describe but not to limit the technical solutions of this application; and within the spirits of this application, technical features of the above embodiments or different embodiments may also be combined with each other, the steps may be implemented in an arbitrary order, and many other variations in different aspects of this application described above are possible although, for purpose of simplicity, they are not provided in the details. Although this application has been detailed with reference to the above embodiments, those of ordinary skill in the art shall appreciate that modifications can still be made to the technical

solutions disclosed in the above embodiments or equivalent substations may be made to some of the technical features, and the corresponding technical solutions will not essentially depart from the scope of the embodiments of this application due to such modifications or substations.

Claims

1. A computing power scheduling platform, being characterized in that, comprising:

an order management unit, being configured to receive at least one computing task from a platform user;

a storage space allocation unit, being configured to allocate a data storage space with a target capacity for the platform user;

a computing power allocation unit, being configured to allocate a corresponding computing node to the computing task according to computing power requirements of the computing task;

an expense calculation unit, being configured to record the usage of computing resources of the computing task and generate corresponding expense information;

wherein the computing task is packaged in an independent container, and the container is launched on the computing node allocated by the computing power allocation unit so that the computing task is executed;

data information formed after the computing task is executed is stored in the data storage space allocated by the storage space allocation unit.
2. The computing power scheduling platform according to Claim 1, being characterized in that, further comprising a monitoring unit which is configured to:

feedback the state information of the computing node during the execution of the computing task; and the state information is selected from one or more of the following:

CPU load; GPU load and physical memory usage ratio.
3. The computing power scheduling platform according to Claim 1, being characterized in that, further comprising: a platform interface presented on a user terminal;

wherein the platform interface comprises an access page for accessing the container.
4. The computing power scheduling platform according to Claim 3, being characterized in that, the platform interface further comprises a user login page for the platform user to register and log in.
5. The computing power scheduling platform according to Claim 1, being characterized in that, the target capacity is determined by the platform user; and the data storage space is mapped into the container.

6. The computing power scheduling platform according to Claim 5, being characterized in that, the data storage space is provided with a plurality of storage space copies.
7. The computing power scheduling platform according to Claim 5, being characterized in that, the data information formed after the computing task is executed is stored by a persistent storage mechanism so that the data information is independent of the life cycle of the container.
8. The computing power scheduling platform according to Claim 3, being characterized in that, the computing power allocation unit is specifically configured to:

determine the computing power requirements of the computing task;

authorize a computing node meeting the computing power requirements to a corresponding platform user, and binding the authorized computing node with the platform user.
9. A computing power leasing system, being characterized in that, comprising:

a physical device providing computing power resources and storage spaces; and

the computing power scheduling platform according to any of Claims 1 to 8;

wherein the computing power scheduling platform abstracts the computing power resources and the storage spaces of the physical device through the container and virtualization technology to form corresponding virtual resources.
10. The computing power leasing system according to Claim 9, being characterized in that, the computing power scheduling platform forms a virtual resource corresponding to the physical device through a Docker container and a KVM virtualization platform.